parameter_boundary

This function is sourced and called throughout the optimization workflow for boundaries. It contains all the parameters I used. MAKE SURE DOCUMENTATION MATCHES THE ACTUAL SET VALUES (make sure listed parameter ranges are reflected in the function)

Functions for parameters

Biological information, parameter interpretation given at the end. Also a calculation to draw ranges of β are shown but the model constraints are set to be much wider.

Boundary Function used in the optimizations – all in $\log_{10}$

```
#### EDIT THIS IN THE DOCUMENTATION FILE (.Rmd) ONLY ####


boundary_set = function(parms, boundary = NULL, t_set = NA, par_set = NA, initV_upper = 2){
  if(length(parms) == 0 | is.null(boundary)) return(NULL)

  #remember these are all log10ed

  if(boundary %in% c("l", "lower", "L", "Lower")){
    set_values = c(
      beta = -14,
      delta = -4,
      theta = -3,
      KI = 0,
      KT = 2,
      death = -5,
      start_day = log10(0.1)
    )
    return_list = sapply(1:length(parms), function(i) which(names(set_values) == names(parms)[i]))

    return(unname(set_values[return_list]))
  }

  if(boundary %in% c("u", "upper", "U", "Upper")){
    set_values = c(
      beta = -6.5,
      delta = 3,
      theta = 2,
      KI = 8,
      KT = 10,
      death = 0,
      start_day = log10(25)
    )
    return_list = sapply(1:length(parms), function(i) which(names(set_values) == names(parms)[i]))

    return(unname(set_values[return_list]))
  }

  print("Incorrect boundary label given (use 'lower' or 'upper')")
  return(NULL)
}
```

Latin hypercube initial value draw functions

R0 checking (constraining the sampling range) is better handled directly in the specific optimization scripts.

```
#this returns a list of a set of initial value draws for given set of parameters
#uses LHS package
#R0check is only for when beta and delta are fit together
draw_initial = function(fitparms, lower, upper,
                        total_draws = 50, save.out = T, R0check = F, file_name = NULL){
  totalFit = length(fitparms)

  if(totalFit != length(lower) | totalFit != length(upper)) {
    print(paste("mismatch, 1=lower, 2=upper:", which(!c(length(lower), length(upper)) %in% totalFit)))
    return(NULL)
  }


  lhs = randomLHS(total_draws, totalFit)

  starting_sets = llply(1:total_draws, function(i){
    tempdraws = (upper - lower) * lhs[i, ] + lower
    names(tempdraws) = names(fitparms)

    if(R0check){
      tempR0 = with(as.list(10^tempdraws), beta * 4e8 * 60 /(delta * 2))
      if(tempR0 < 0.5) return(NULL)
    }

    tempdraws
  })

  starting_sets <- compact(starting_sets) #compact removes NULLs (remove nothing if R0check = F)
  total_kept = length(starting_sets) #should be same as total_draw when R0check = F

  if(save.out){
    starting_sets_save = ldply(1:total_kept, function(i) starting_sets[[i]])
    if(is.null(file_name)) file_name = Sys.Date()
    write.csv(starting_sets_save,
              paste(file_name,".csv", sep = ""),
              row.names = F)
  }

  return(starting_sets)

}
```

addnames_fit

```r
#this tacks on parameters names that werent used in the model for rbinding in make_plot_data and main l
addnames_fit = function(output, parmnames){
  addnames = which(!parmnames %in% names(output))

  if(length(addnames) == 0) return(output)

  temp_output = cbind(output, t(rep(NA, length(addnames))))
  names(temp_output) = c(names(output), parmnames[addnames])

  return(temp_output)

}
```

Background information

Equations

1. SIV model with no immunity

$$
\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= \lambda - \mu S - \beta S V \\
\frac{\mathrm{d}I0}{\mathrm{d}t} &= \beta S V - \alpha I0 - \mu I0 \\
\frac{\mathrm{d}I}{\mathrm{d}t} &= \alpha I0 - \delta I - k I T \\
\frac{\mathrm{d}V}{\mathrm{d}t} &= p I - c V \\
\frac{\mathrm{d}T}{\mathrm{d}t} &= \theta \frac{I}{K_I + I} - death * T
\end{aligned}
$$

2. Model with cytolytic immunity

$$
\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= \lambda - \mu S - \beta S V \\
\frac{\mathrm{d}I0}{\mathrm{d}t} &= \beta S V - \alpha I0 - \mu I0 \\
\frac{\mathrm{d}I}{\mathrm{d}t} &= \alpha I0 - \delta I - k I T \\
\frac{\mathrm{d}V}{\mathrm{d}t} &= p I - c V \\
\frac{\mathrm{d}T}{\mathrm{d}t} &= \theta \frac{I}{K_I + I} - death * T
\end{aligned}
$$

3. **Model with viral-mediated response**

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \lambda - \mu S - \beta S V$$
$$\frac{\mathrm{d}I0}{\mathrm{d}t} = \beta S V - \alpha I0 - \mu I0$$
$$\frac{\mathrm{d}I}{\mathrm{d}t} = \alpha I0 - \delta I$$
$$\frac{\mathrm{d}V}{\mathrm{d}t} = pI - cV - kTV$$
$$\frac{\mathrm{d}T}{\mathrm{d}t} = \theta \frac{V}{K_T + V} - death * T$$

Biologically fixed parameter interpretation and ranges

$\lambda = K\mu$ – Fixed growth rate of S fixed

$K = 10^7 * 40$ Fixed total S size (from literature, Dawes 2003)

$\mu = 1/4.5$ – Death rate of S or I0 (also from Dawes)

$\beta \in 10^{-14}, 10^{-6.5}$ – Infectivity, see below for constraint calculations.

$\alpha = 1$ – Latency period before virus replicates

$\delta = 0.77$ – Death rate of I, per day (Emery paper)

$\delta \in (10^{-4}, 10^3)$ – These boundaries were used when profiling over μ.

$p = 1600$ – PNAS temperature paper.

$c = 2$ – Viral clearance rate (generally fixed), 2 from EBV model paper

$k = 0.01$ – Immune clearance of virus (or maybe I), fixed from several papers at 1% per effectory per day.

$\theta \in (0.001, 100)$ – Immune activation rate

$K_I \in (100, 10^8)$ – Infected cell population where immune system is 50% activated

$death \in (0.00001, 1)$ – Death rate of immune effector population

$K_T \in (100, 10^{10})$ – Viral level where immune system is 50% activated

start_day or $t_0 \in (0.1, 25)$ – Days before detection of first positive. The day when $I_0 = 1$ in the model simulation.

Constraining β for sampling

$$R_0 = \frac{\beta p \lambda}{c\delta\mu(1 + \frac{\mu}{\alpha})} = \frac{\beta p K}{c\delta(1 + \frac{\mu}{\alpha})}$$

$$1 < R_0 < X$$

so the general solution for β here is

$$\frac{c\delta(1 + \frac{\mu}{\alpha})}{pK} < \beta < X * \frac{c\delta(1 + \frac{\mu}{\alpha})}{pK}$$

β was sampled from this range for target cell model. When μ was varied and δ optimized with β, LHS sampled values were tested to make sure R0 was between 1 and 50.